

ZABBIX '24

CONFERENCE

LATIN AMERICA

JUNE, 07-08, 2024. SÃO PAULO, BRAZIL

ZABBIX '24

CONFERENCE

LATIN AMERICA

TRABAJANDO CON LLDs: Todo lo que necesitas saber

ABOUT ME:



- Trainer LATAM
- Support TEAM
- Networker / SysAdmin
- Padre / Moto-fan

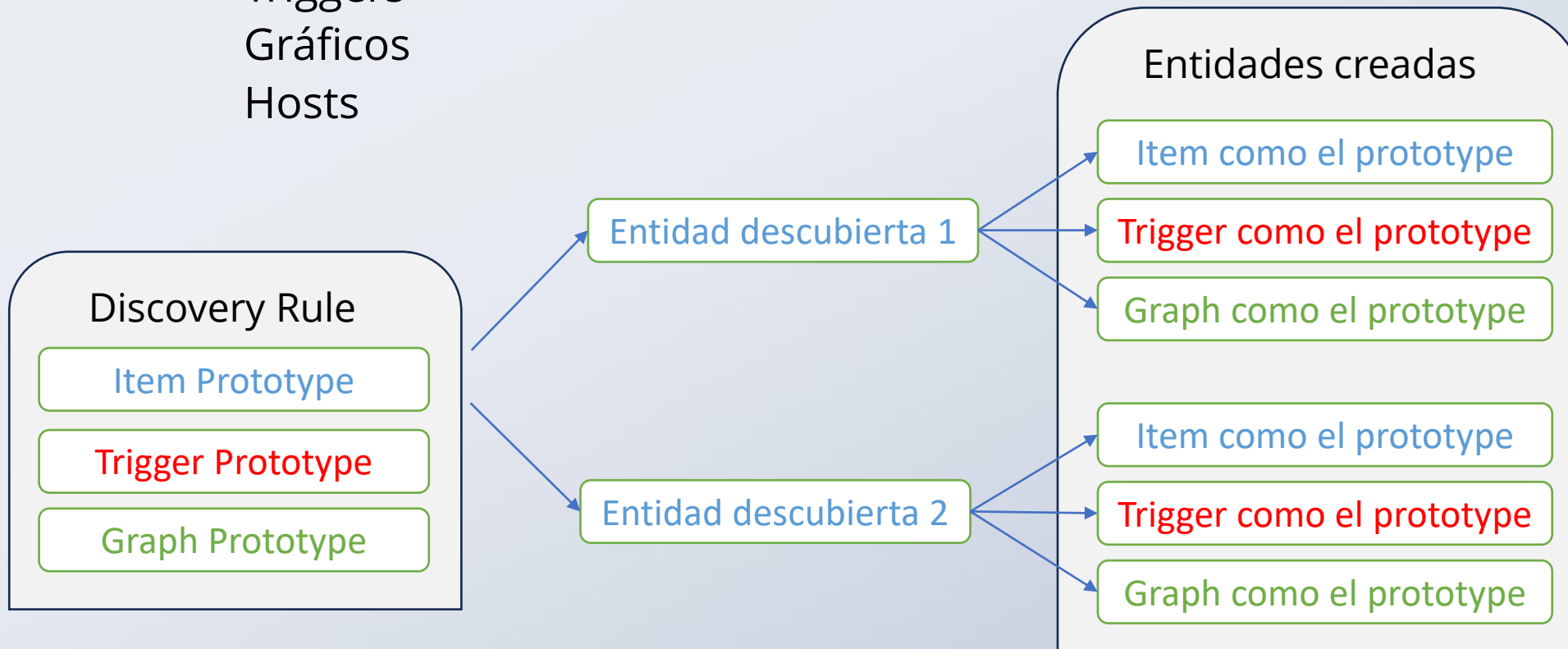


Que es Low-Level Discovery?

LLD es una funcionalidad de **Zabbix** para automatizar la creación de **entidades**.

Simplifica la gestión de entornos de gran escala y dinámicos creando:

Items
Triggers
Gráficos
Hosts



ZABBIX '24

CONFERENCE

LATIN AMERICA

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

El *tipo* define como será alimentada de información nuestra regla de descubrimiento

⚡ Métodos similares a un ítem regular

Nos permite establecer **intervalos** de actualización

* Update interval

Custom intervals

Type	Interval	Period
Flexible Scheduling	50s	1-7,00:00-24:00

Podemos definir **tiempos de retención**

* Delete lost resources ? Never Immediately After

* Disable lost resources ? Never Immediately After

Dependent item ▾

- Zabbix agent
- Zabbix agent (active)
- Simple check
- SNMP agent
- Zabbix internal
- Zabbix trapper
- External check
- Database monitor
- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- JMX agent
- Dependent item
- Script

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

El *tipo* define como será alimentada de información nuestra regla de descubrimiento

⚡ Métodos similares a un ítem regular

Nos permite establecer intervalos de actualización

```
[root@zbx70 ~]# zabbix_agentd -t net.if.discovery
net.if.discovery [s] [{"#IFNAME":"lo"}, {"#IFNAME":"eth0"}, {"#IFNAME":"br-2a8056d33cd1"}, {"#IFNAME":"docker0"}]]
[root@zbx70 ~]#
```

```
[root@zbx70 ~]# zabbix_agentd -t vfs.fs.get
vfs.fs.get [s] [{"fsname":"/", "fstype":"ext4", "bytes":{"total":18843648000, "free":5900013568, "pfree":86.621772, "pused":13.378228}, "options":"rw,relatime,stripe=16"}, {"fsname":"/dev", "fstype":"tmpfs", "bytes":{"total":5038080, "free":5038080, "pfree":99.997505, "pused":0.002495}, "options":"rw,relatime,size=492k,mode=755,uid=100000,gid=100000,inode64"}, {"fsname":"/proc/sys/net/ipv4/tcp_max_syn_backlog", "fstype":"proc", "bytes":{"total":0, "free":0, "used":0, "pfree":100.000000, "pused":0.000000}, "options":"rw,nosuid,nodev,noexec,relatime"}, {"fsname":"/proc/sys/net/ipv4/tcp_max_tw_buckets", "fstype":"proc", "bytes":{"total":0, "free":0, "used":0, "pfree":100.000000, "pused":0.000000}, "options":"rw,nosuid,nodev,noexec,relatime"}, {"fsname":"/proc/sys/net/ipv4/tcp_syncookies", "fstype":"proc", "bytes":{"total":0, "free":0, "used":0, "pfree":100.000000, "pused":0.000000}, "options":"ro,relatime"}, {"fsname":"/proc/sysrq-trigger", "fstype":"proc", "bytes":{"total":0, "free":0, "used":0, "pfree":100.000000, "pused":0.000000}, "options":"ro,relatime"}]]
```

Dependent item

Zabbix agent

Zabbix internal

SSH agent

TELNET agent

JMX agent

Dependent item

Script

* Delete lost resources ? Never Immediately **After** 7d

* Disable lost resources ? Never **Immediately** After

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

Item Tags Preprocessing

* Name

Type

* Key

Type of information

* Host interface

* SNMP OID

- Dependent item
- Zabbix agent
 - Zabbix agent (active)
 - SNMP agent**
 - Zabbix internal
 - Zabbix trapper
 - External check
 - Database monitor
 - HTTP agent
 - IPMI agent
 - SSH agent
 - TELNET agent
 - JMX agent
 - Dependent item
 - Script

Podemos definir tiempos de actualización

```
.1.3.6.1.2.1.2.2.1.2.1 = STRING: "lo"  
.1.3.6.1.2.1.2.2.1.2.2 = STRING: "ens33"  
.1.3.6.1.2.1.2.2.1.2.3 = STRING: "ens37"  
.1.3.6.1.2.1.2.2.1.3.1 = INTEGER: 24  
.1.3.6.1.2.1.2.2.1.3.2 = INTEGER: 6  
.1.3.6.1.2.1.2.2.1.3.3 = INTEGER: 6
```

* Delete lost resources

* Disable lost resources

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

Discovery rule Preprocessing LLD macros Filters Overrides

* Name Proxy discovery

Type Database monitor

* Key db.odbc.discovery[proxies,{\$DSN}]

User name

Password

* SQL query
SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxy_hostid WHERE h1.status IN (5, 6) GROUP BY h1.host;

formación nuestra regla de

realización

- Dependent item
- Zabbix agent
- Zabbix agent (active)
- Simple check
- SNMP agent
- Zabbix internal
- Zabbix trapper

Database monitor

Podemos definir tiempos de retención

* Delete lost resources ? Never Immedia

* Disable lost resources ? Never Immedia

```
[  
{  
  "#{#HOST}": "Japan 1",  
  "#{#COUNT}": "5"  
},  
{  
  "#{#HOST}": "Japan 2",  
  "#{#COUNT}": "12"  
},  
]
```

- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- JMX agent
- Dependent item
- Script

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

Discovery rule Preprocessing LLD macros 2 Filters 2 Overrides

Parent discovery rules AWS by HTTP

* Name EC2 instances discovery

Type Script

* Key aws.ec2.discovery

Parameters	Name	Value	Action
	access_key	{AWS.ACCESS.KEY.ID}	Remove
	proxy	{AWS.PROXY}	Remove
	region		
	secret_key		
	Add		

* Script var AwsEC2 = {...

de información nuestra regla de

de actualización

JavaScript

```
122  
123     return instance;  
124   },  
125  
126   listInstances: function () {  
127     var payload = {};  
128  
129     payload['Action'] = 'DescribeInstances',  
130     payload['Version'] = '2016-11-15'  
131
```

Dependent item

- Zabbix agent
- Zabbix agent (active)
- Simple check
- SNMP agent
- Zabbix internal
- Zabbix trapper
- External check
- Database monitor
- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- JMX agent

Podemos definir tier

* Delete lost resource

* Disable lost resource

Script

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

El resultado del **intervalo** más la política de **retención** definen el comportamiento de las entidades descubiertas frente a cambios:

Dependent item	<u>Disabled</u>	owner: Leonardo Luis ...	<i>i</i>
Dependent item	<u>Disabled</u>	owner: Leonardo Luis ...	
Dependent item	<u>Disabled</u>	owner: Victor Breda C...	
Dependent item	<u>Disabled</u>	owner: Victor Breda C...	<i>i</i>
Dependent item	<u>Disabled</u>	owner: Victor Breda C...	

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

Utilizando preprocesamiento para obtener un JSON

📶 Ejemplo de datos sin procesar:

TABLE_NAME	DATA_LENGTH	INDEX_LENGTH
acknowledges	16384	49152
actions	16384	32768

📶 Podemos utilizar pasos de preprocesamiento para preparar los datos para la regla LLD, transformando esos datos en JSON

Preprocessing steps	Name	Parameters
1:	Replace	<input type="text" value="\t"/> ;
2:	CSV to JSON	<input type="text" value=";"/> <input type="text" value=""/> <input checked="" type="checkbox"/> With head

Después del preprocesamiento, los datos JSON están listos para usarse en LLD

```
[{  
  "TABLE_NAME": "acknowledges",  
},  
{  
  "TABLE_NAME": "actions",  
}]
```

DISCOVERY
TYPE

PRE
PROCESSING

LLD MACRO

FILTERS

OVERRIDE

Para JSON personalizados, es necesario realizar un mapeo utilizando JSONPath

⚡ Este mapeo es realizado en las propiedades de la regla LLD, en la solapa LLD macros

```
{  
  "fsname": "/sys",  
  "fstype": "sysfs",  
  "bytes": {"total": 941412352, "free": 941412352}  
}, {  
  "fsname": "/var",  
  "fstype": "xfs",  
  "bytes": {"total": 53673439232, "free":  
45409468416}  
}
```



LLD macros	LLD macro	JSONPath	
	{#FS.NAME}	\$.fsname	Remove
	{#FS.TYPE}	\$.fstype	Remove

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

La solapa *filtro* es utilizada para permitir solo entidades que cumplan un criterio determinado:

- ✓ Expresiones regulares (PCRE regex)
- ✓ User macro (Host, Template, Global)
- ✓ Global regular expression

The screenshot displays the Zabbix configuration interface for a discovery rule. The 'Filters' tab is active, showing a configuration for a discovery rule with the following settings:

- Type of calculation:** And (A and B) and (C and D)
- Filters:**
 - Filter A:** LabelMacro: `{#FSNAME}`, Regular expression: `{$VFS.FS.FSNAME.MATCHES}`, Action: matches
 - Filter B:** LabelMacro: `{#FSNAME}`, Regular expression: `{$VFS.FS.FSNAME.NOT_MATCHES}`, Action: does not match

Below the filters, the 'Macros' tab is active, showing the following macro mappings:

Macro	Value	Description
<code>{\$VFS.FS.FSNAME.MATCHES}</code>	<code>.+</code>	Filesystem names to discover
<code>{\$VFS.FS.FSNAME.NOT_MATCHES}</code>	<code>^(/dev sys run proc .+/shm\$)</code>	Filesystem names not to discover

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

Los overrides permiten definir reglas las cuales modifican atributos de entidades descubiertas:

- ✓ Son mostradas en una lista de prioridades modificable
- ✓ Las reglas utilizan macros de descubrimiento para aplicar sus criterios

Discovery rule Preprocessing LLD macros Filters Overrides

Overrides

Name	Stop processing	Action
1: Trigger severity for trunk port	Yes	Remove
2: Trigger severity for server port	Yes	Remove
3: Trigger severity for regular port	Yes	Remove
4: Not monitored ports	No	Remove

[Add](#)

[Update](#) [Clone](#) [Test](#) [Delete](#) [Cancel](#)

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

Definimos el nombre de la regla de override:

- ⚡ El campo de filtro es verificado contra la macro LLD :
 - ✓ Utiliza una expresión regular
 - ✓ Verifica si la macro este definida en primer lugar

- ⚡ Es posible realizar múltiples operaciones si el filtro es satisfecho:

Override

* Name

If filter matches

Filters	Label	Macro		Regular expression	Action
	A	<input type="text" value="{#IFNAME}"/>	matches	<input type="text" value="trunk"/>	Remove

[Add](#)

Operations	Condition	
	Trigger prototype equals <i>Interface</i> <i>{#IFNAME}</i> : Link down	Edit Remove

[Add](#)

DISCOVERY
TYPE

PRE
PROCESSING

LLD
MACRO

FILTERS

OVERRIDE

Operaciones:

⚡ Afecta entidades creadas por la LLD :

- ✓ Items
- ✓ Triggers
- ✓ Graficos
- ✓ Hosts creados vía prototypes

⚡ Condition – Verifica el nombre de la entidad

- ✓ Equals / Does not equal
- ✓ Contains / Does not contains
- ✓ Matches / Does not matches

⚡ Cambia los atributos de objetos descubiertos dado un criterio:

- ✓ Create enabled: (Yes/No)
- ✓ Discover (Yes/No)
- ✓ Otros atributos de la entidad (History/trends/severity/etc)

New operation

Object

Condition

Create enabled Yes No

Discover Yes No

Update interval Delay

Custom intervals

Type	Interval	Period	Action
Add			

History storage period Original

Trend storage period

Tags Original

Type of information

Units

* Update interval

Custom intervals

Type	Interval	Period	Action
Flexible Scheduling	50s	1-7,00:00-24:00	Remove
Add			

PROTOTYPES

* History storage period Storage period

* Trend storage period Storage period

Value mapping

Description

Create enabled

Discover

ITEM PROTOTYPE

Los prototipos de items tienen las mismas propiedades que los items normales

Requisitos obligatorios:

LLD `{#MACRO}` en la key del item

La key del item debe ser **única**

Utilizar `{#MACRO}` como [parámetro] de la key

Mejor practica (recomendado):

Utilizar `{#MACRO}` como nombre del ítem

Utilizar `{#MACRO}` en TAGs

Item prototype Tags 1 Preprocessing 2

* Name Interface `{#IFNAME}`: Bits received

Type Zabbix agent (active)

* Key net.if.in["`{#IFNAME}`"]

Type of information

Units

* Update interval

Custom intervals

* Name Interface `{#IFNAME}`: Bits received

Type Zabbix agent (active)

* Key net.if.in["`{#IFNAME}`"]

* History storage period Do not keep history Storage period 7d

* Trend storage period Do not keep trends Storage period 365d

Value mapping type here to search

Description

Create enabled

Discover

Add Test Cancel

Item prototypes 3 Trigger prototypes 2 Graph prototypes 1 Host prototypes

<input type="checkbox"/>	Name ▲	Key	Interval
<input type="checkbox"/>	... <code>{#FSNAME}</code> : Space utilization	vfs.fs.size <code>{#FSNAME}</code> ,used]	1m
<input type="checkbox"/>	... <code>{#FSNAME}</code> : Total space	vfs.fs.size <code>{#FSNAME}</code> ,total]	1m
<input type="checkbox"/>	... <code>{#FSNAME}</code> : Used space	vfs.fs.size <code>{#FSNAME}</code> ,used]	1m

TRIGGER PROTOTYPE

Recomendado utilizar LLD `{#MACRO}` en nombre del trigger:

LLD `{#MACRO}` puede ser usando en tags

⚡ Expresión del trigger puede contener:

- ✓ Item prototype de la misma regla LLD
- ✓ Item normales del cualquier host/template
- ✓ Lo mismo vale para dependencias de triggers

Condition dialog box showing configuration for a trigger condition. Fields include: Item (with Select and Select prototype buttons), Function (last() - Last (most recent) T value), Last of (T) (with Count), Time shift (with Time), and Result (= 0). Buttons: Insert, Cancel.

Trigger prototype configuration page. Fields include: Name, Event name, Operational data, Severity (Warning selected), and Expression. Expression: last(/Linux filesystems by Zabbix agent active/vfs.fs.size[{#FSNAME},pused])>{\$VFS.FS.PUSED.MAX.WARN:"{#FSNAME}"}. Buttons: Add, Expression constructor.

<input type="checkbox"/>	Warning	Free disk space is less than 20% on volume {#FSNAME}	Space used: {ITEM.LASTVALUE1}	last(/Linux by Zabbix agent/vfs.fs.size[{#FSNAME},pused])>80
<input type="checkbox"/>	Warning	Free inodes is less than 20% on volume {#FSNAME}	Free inodes: {ITEM.LASTVALUE1}	min(/Linux by Zabbix agent/vfs.fs.inode[{#FSNAME},pfree],5m)<20

RESULTADO

Una vez creadas las entidades basadas en la información de la regla y en las definiciones de prototipos, encontraremos prefijos:

⚡ El prefijo indica la regla responsable de la creación

✓ Para items:

Mounted filesystem discovery: FS [/]: Get data: FS [/]: Space: Used, in %	Triggers 2	vfs.fs.dependent.size[/,pused]
Mounted filesystem discovery: FS [/]: Get data: FS [/]: Inodes: Free, in %	Triggers 2	vfs.fs.dependent.inode[/,pfree]
Network interface discovery: Interface eth0: Inbound packets discarded		net.if.in["eth0",dropped]
Network interface discovery: Interface eth0: Inbound packets with errors	Triggers 1	net.if.in["eth0",errors]

✓ Para triggers:

<input type="checkbox"/>	Information	OK	Linux by Zabbix agent: Configured max number of processes is too low Depends on: Zabbix server: Getting closer to process limit	<code>last(/Zabbix server/kernel.maxproc)<{\$KERNEL.MAXPROC.MIN}</code>
<input type="checkbox"/>	Average	OK	Mounted filesystem discovery: FS [/]: Filesystem has become read-only	Problem: <code>last(/Zabbix server/vfs.fs.dependent[/,readonly],#2)=0</code> and <code>last(/Zabbix server/vfs.fs.dependent[/,readonly])=1</code> Recovery: <code>last(/Zabbix server/vfs.fs.dependent[/,readonly])=0</code>
<input type="checkbox"/>	Average	OK	Mounted filesystem discovery: FS [/]: Running out of free inodes	<code>min(/Zabbix server/vfs.fs.dependent.inode[/,pfree],5m)<{\$VFS.FS.INODE.PFREE.MIN.CRIT:"/"}</code>

PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

- ⚡ La información es obtenida por los recolectores de información:
 - ✓ POLLERS
 - ✓ TRAPPERS
- ⚡ La información puede ser procesada (El resultado debe ser JSON)
- ⚡ Esta información es enviada al **LLD Manager** para asignar a **Workers**

Name ▲	Last check	Last value	Change
LLD queue ?	2s	0	
Utilization of LLD manager internal processes, in % ?	1m	0.0169 %	+0.0169 %
Utilization of LLD worker internal processes, in % ?	58s	0.1436 %	+0.006101 %

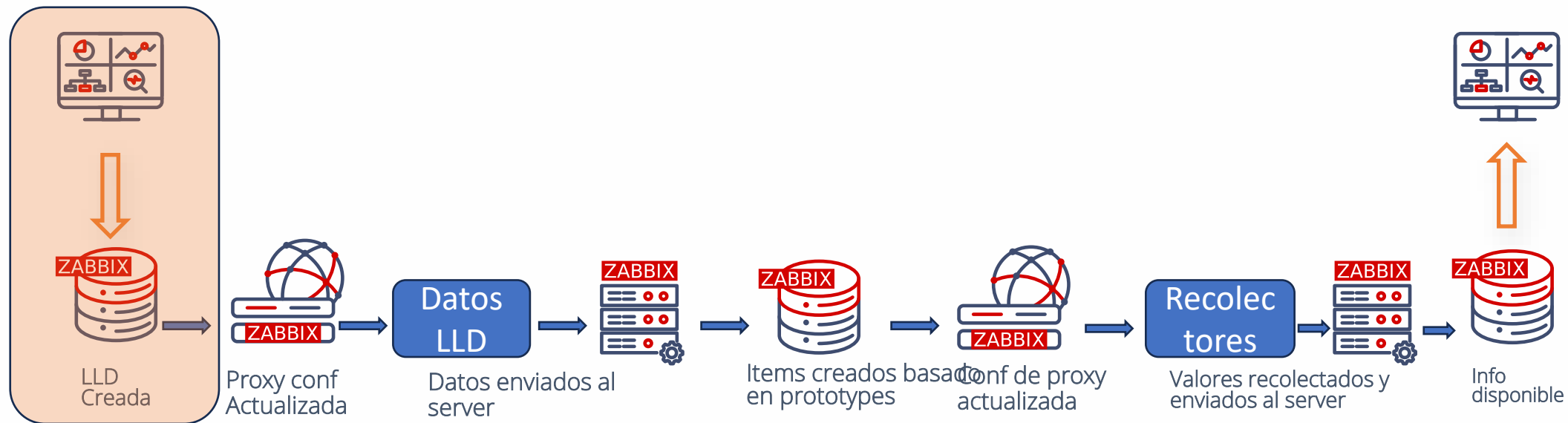
```
### Option: StartLLDProcessors
#   Number of pre-forked instances of low level discovery processors.
#
# Mandatory: no
# Range: 1-100
# Default:
# StartLLDProcessors=2
```

```
zabbix 276549 276525 0 May14 ? 00:01:22 /usr/sbin/zabbix_server: lld manager #1 [processed 0 LLD rules, idle 5.019072sec during 5.019164 sec]
zabbix 276550 276525 0 May14 ? 00:01:46 /usr/sbin/zabbix_server: lld worker #1 [processed 1 LLD rules, idle 36.306665 sec during 36.313753 sec]
zabbix 276551 276525 0 May14 ? 00:01:20 /usr/sbin/zabbix_server: lld worker #2 [processed 1 LLD rules, idle 51.141062 sec during 51.148535 sec]
```

PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

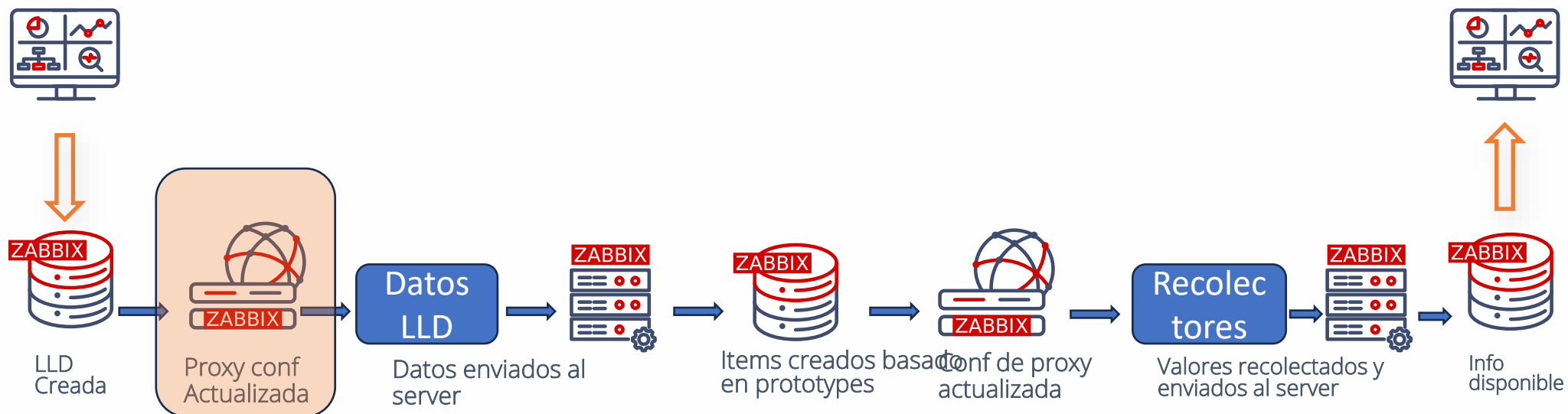
- ⚡ La regla LLD es creada en la DB del server:
- ⚡ El proxy recibe la configuración de la regla LLD
- ⚡ El proxy procesa la regla y envía la información de LLD al server
- ⚡ Server procesa la información LLD y crea las entidades
- ⚡ El proxy actualiza su configuración con las nuevas entidades.



PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

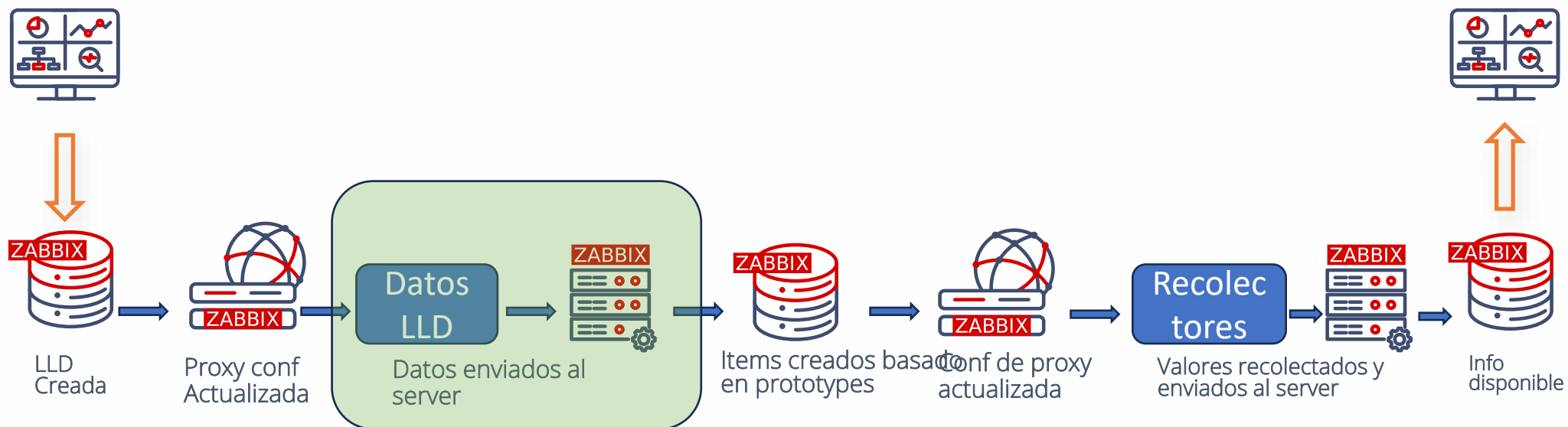
- ⚡ La regla LLD es creada en la DB del server:
- ⚡ El proxy recibe la configuración de la regla LLD
- ⚡ El proxy procesa la regla y envía la información de LLD al server
- ⚡ Server procesa la información LLD y crea las entidades
- ⚡ El proxy actualiza su configuración con las nuevas entidades.



PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

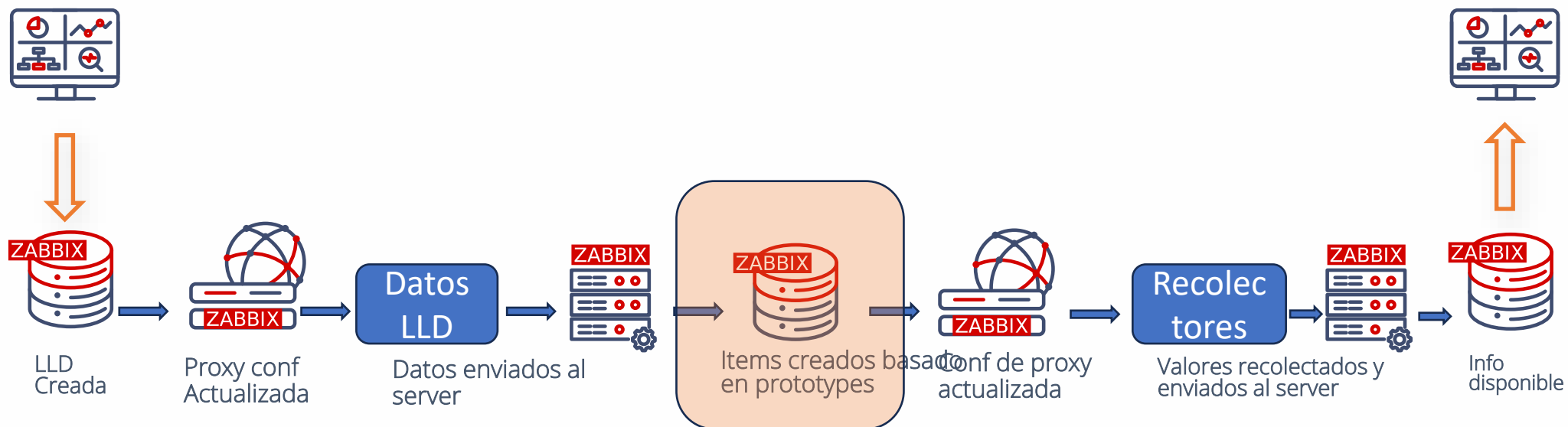
- ⚡ La regla LLD es creada en la DB del server:
- ⚡ El proxy recibe la configuración de la regla LLD
- ⚡ El proxy procesa la regla y envía la información de LLD al server
- ⚡ Server procesa la información LLD y crea las entidades
- ⚡ El proxy actualiza su configuración con las nuevas entidades.



PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

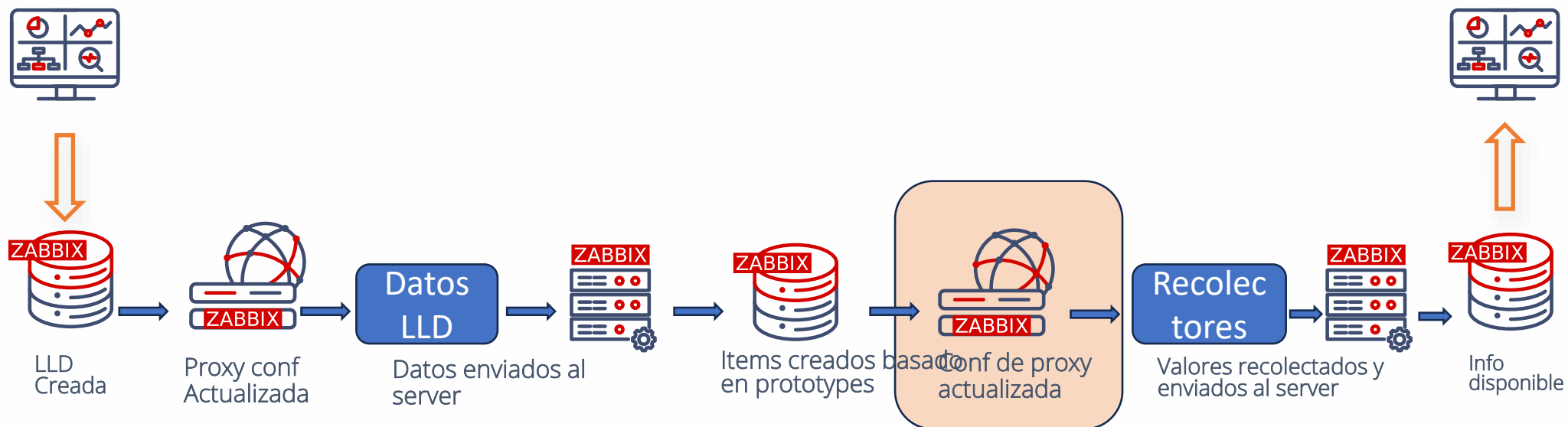
- ⚡ La regla LLD es creada en la DB del server:
- ⚡ El proxy recibe la configuración de la regla LLD
- ⚡ El proxy procesa la regla y envía la información de LLD al server
- ⚡ Server procesa la información LLD y crea las entidades
- ⚡ El proxy actualiza su configuración con las nuevas entidades.



PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

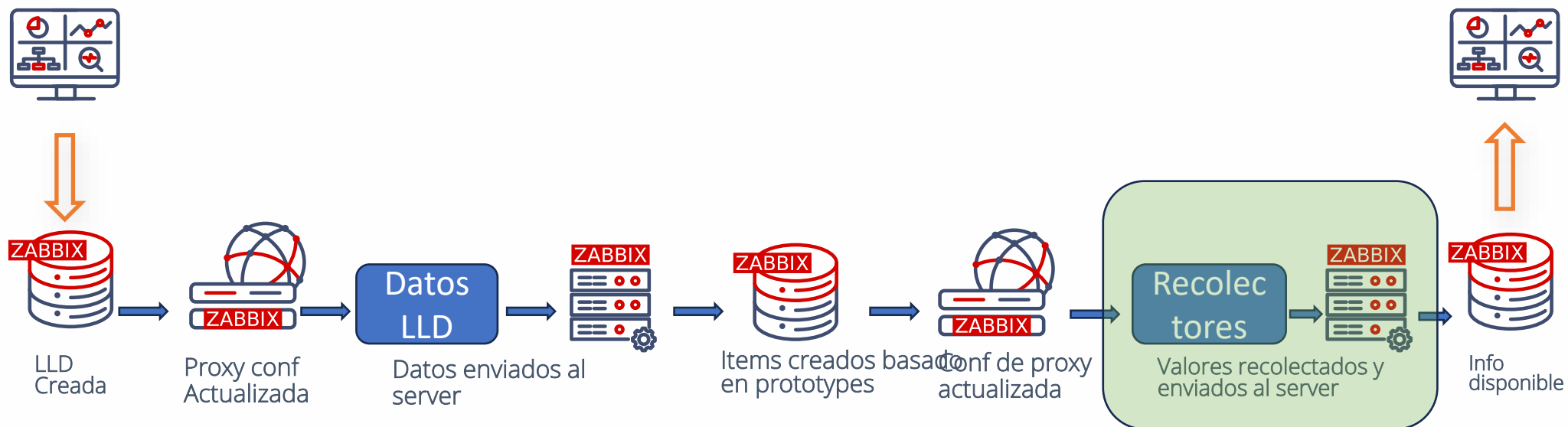
- ⚡ La regla LLD es creada en la DB del server:
- ⚡ El proxy recibe la configuración de la regla LLD
- ⚡ El proxy procesa la regla y envía la información de LLD al server
- ⚡ Server procesa la información LLD y crea las entidades
- ⚡ El proxy actualiza su configuración con las nuevas entidades.



PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

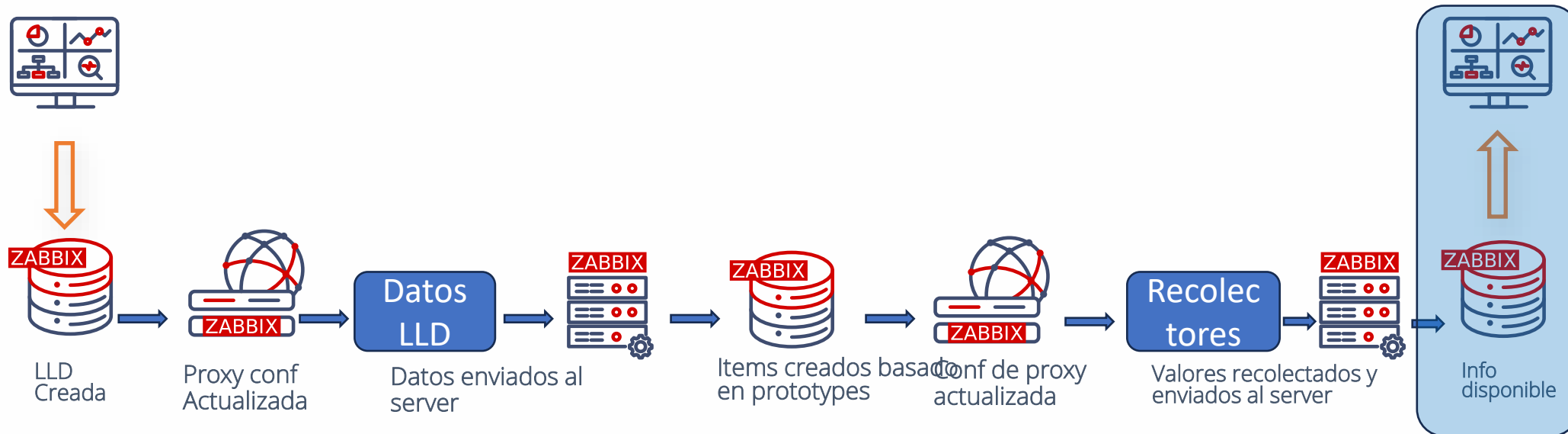
- ⚡ La regla LLD es creada en la DB del server:
- ⚡ El proxy recibe la configuración de la regla LLD
- ⚡ El proxy procesa la regla y envía la información de LLD al server
- ⚡ Server procesa la información LLD y crea las entidades
- ⚡ El proxy actualiza su configuración con las nuevas entidades.



PROCESO INTERNO

El descubrimiento en Zabbix involucra múltiples procesos:

- ⚡ La regla LLD es creada en la DB del server:
- ⚡ El proxy recibe la configuración de la regla LLD
- ⚡ El proxy procesa la regla y envía la información de LLD al server
- ⚡ Server procesa la información LLD y crea las entidades
- ⚡ El proxy actualiza su configuración con las nuevas entidades.



¡Gracias!

Obrigado!

Thank you!

